

Program listing for Cash Register (Tasks 1 to 5)

```
unit main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    edtCost: TEdit;
    edtPrice: TEdit;
    edtQuantity: TEdit;
    lblQuantity: TLabel;
    lblPrice: TLabel;
    lblCost: TLabel;
    btnEnter: TButton;
    edtVAT: TEdit;
    edtTotalCost: TEdit;
    lblVAT: TLabel;
    Label3: TLabel;
    edtTransaction: TButton;

    procedure btnEnterClick(Sender: TObject);
    procedure edtTransactionClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }

    rlMemoryCost: real;
    rlMemoryVAT : real;
    rlMemoryTotalCost : real;

  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.btnEnterClick(Sender: TObject);
var
  intQuantity: integer;
  rlPrice, rlCost, rlVat, rlTotalCost: real;
  strCost, StrVat, StrTotalCost: string;
  ErrorA, ErrorB: integer;
begin
  rlTotalCost := 0.0;
  Val(edtQuantity.Text, intQuantity, ErrorA);
  Val(edtPrice.Text, rlPrice, ErrorB);
  if (edtQuantity.Text = '') and (edtPrice.Text = '') then
    ShowMessage('Quantity and Price cannot both be blank')
  else
```

```

begin
    if (ErrorA <> 0) and (ErrorB <> 0) then
        ShowMessage('Quantity and Price are both be Invalid!! -
Please try again!')
    else if edtQuantity.Text = '' then
        ShowMessage('The edit box for Quantity is blank!! - Enter a
value for Quantity')
    else if edtPrice.Text = '' then
        ShowMessage('The edit box for Price is blank!! - Enter a
value for Price')
    else if ErrorA <> 0 then
        ShowMessage('Please try again! - Invalid Quantity')
    else if ErrorB <> 0 then
        ShowMessage('Please try again! - Invalid Price')
    else if (intQuantity < 0) and (rlPrice < 0) then
        ShowMessage('Quantity and Price cannot both be negative')
    else
        begin
            Val(edtQuantity.Text, intQuantity, errorA);
            Val(edtPrice.Text, rlPrice, errorB);
            rlCost := intQuantity * rlPrice;

            rlVat := rlCost*0.175;
            Str(rlVat:0:2, strVat);
            edtVat.Text := StrVat;
            rlTotalCost:= rlVat + rlCost;
            Str(rlTotalCost:0:2, StrTotalCost);
            edtTotalCost.Text :=StrTotalCost;
            Str(rlCost:0:2, strCost);
            edtCost.Text := strCost;

            rlMemoryTotalCost :=rlMemoryTotalCost + rlTotalCost;
            rlMemoryCost := rlMemoryCost + rlCost;
            rlMemoryVAT := rlMemoryVAT + rlVAT;

        end;
    end;
end;

procedure TForm1.edtTransactionClick(Sender: TObject);
var
    strMemory, strMemoryCost, strMemoryCash : string;
    strMemoryChange, strMemoryVAT, strMemoryTotalCost: string;
    rlTotal : real;
begin
    str(rlMemoryCost:0:2,strMemoryCost);
    edtCost.Text :=strMemoryCost;

    str(rlMemoryVAT:0:2,strMemoryVAT);
    edtVAT.Text :=strMemoryVAT;

    str(rlMemoryTotalCost:0:2,strMemoryTotalCost);
    edtTotalCost.Text :=strMemoryTotalCost;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    rlMemoryCost:=0;
    rlMemoryVAT :=0;

```

```

    rlMemoryTotalCost :=0;
end;
end.

```

Test Plan for Cash Register Program

Test blank entries:

edtQuantity.Text blank	edtPrice.Text blank
edtQuantity.Text blank	edtPrice.Text non-blank
edtQuantity.Text non-blank	edtPrice.Text blank
edtQuantity.Text non-blank	edtPrice.Text non-blank

Test invalid numeric values:

Test Number format:

edtQuantity.Text integer	ErrorA
edtQuantity.Text real	ErrorA
edtPrice.Text integer	ErrorB
edtPrice.Text real	ErrorB

Test negative values for quantity and price:

Quantity ≥ 0 , price ≥ 0	positive cost calculated
Quantity < 0 , price ≥ 0	negative cost calculated
Quantity ≥ 0 , price < 0	negative cost calculated
Quantity < 0 , price < 0	error message displayed

- Test validation (using the Enter button)

Quantity	Price	Reason	Cost (Ex VAT)	VAT@17.5	Total Cost	Message
Blank	Blank	Test blank entries	Not changed	Not changed	Not changed	Quantity and Price cannot both be blank
blank	Numeric	Testing blank Quantity	Not changed	Not changed	Not changed	The edit box for Quantity is blank!! – Enter a value for Quantity
Numeric	Blank	Test blank Price	Not changed	Not changed	Not changed	The edit box for Price is blank!! –

						Enter a value for Price
Jen	Ken	Invalid Quantity & Price	Not changed	Not changed	Not changed	Quantity and Price are both be Invalid!! - Please try again!
Jjj	12.3	Invalid Quantity	Not changed	Not changed	Not changed	Please try again! - Invalid Quantity
123	Jjj	Invalid Price	Not changed	Not changed	Not changed	Please try again! - Invalid Price
1.85	22.99	Invalid real Quantity	Not changed	Not changed	Not changed	Please try again! - Invalid Quantity
19	1.85	Fractional Price	35.15	6.15	41.30	
2	85	Integer Price	170.00	29.75	199.75	

- Test negative values (without using the transaction button)

Quantity	Price	Reason	Cost (Ex VAT)	VAT@17.5	Total Cost	Message
19	1.85	Negative Price	-35.15	-6.15	-41.30	
-2	85	Negative Quantity	-170.00	-29.75	-199.75	
-19	-19	Negative Quantity & Price	Not changed	Not changed	Not changed	Quantity and Price cannot both be negative

- Test negative values (using the transaction button)

Quantity	Price	Reason	Cost (Ex VAT)	VAT@17.5	Total Cost	Message
19	-1.85	Negative Price	0.00	0.00	0.00	
-2	85	Negative Quantity	0.00	0.00	0.00	
-19	-19	Negative Quantity & Price	0.00	0.00	0.00	

Program listing for Cash Register (Tasks 6 to 8)

```
unit main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    edtNetCost: TEdit;
    edtPrice: TEdit;
    edtQuantity: TEdit;
    lblQuantity: TLabel;
    lblPrice: TLabel;
    lblNetCost: TLabel;
    btnEnter: TButton;
    edtVAT: TEdit;
    edtGrossCost: TEdit;
    lblVAT: TLabel;
    lblGrossCost: TLabel;
    btnClear: TButton;
    btnNext: TButton;
    lblNext: TLabel;
    btnTransaction: TButton;
    btnChange: TButton;
    lblThank: TLabel;
    lblAnd: TLabel;
    edtCash: TEdit;
    edtChange: TEdit;
    lblCash: TLabel;
    lblChange: TLabel;
    Label1: TLabel;
    edtTotal: TEdit;
    lblTotal: TLabel;
    btnTotal: TButton;

    procedure btnEnterClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnNextClick(Sender: TObject);
    procedure edtQuantityChange(Sender: TObject);
    procedure btnTransactionClick(Sender: TObject);
    procedure edtPriceChange(Sender: TObject);
    procedure edtNetCostChange(Sender: TObject);
    procedure edtVATChange(Sender: TObject);
    procedure edtGrossCostChange(Sender: TObject);
    procedure btnChangeClick(Sender: TObject);
    procedure edtCashChange(Sender: TObject);
    procedure btnTotalClick(Sender: TObject);
    procedure edtChangeChange(Sender: TObject);

  private
    { Private declarations }

    rlMemory : real;
    rlMemoryNetCost: real;
```

```

    rlMemoryVAT : real;
    rlMemoryGrossCost : real;
    rlMemoryCash, rlMemoryChange : real;

public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.btnEnterClick(Sender: TObject);
var
    intQuantity: integer;
    rlPrice, rlChange, rlCash, rlCost, rlVat, rlGrossCost: real;
    strCost, StrVat, StrGrossCost: string;
    ErrorA, ErrorB : integer;
begin
    rlGrossCost := 0.0;
    Val(edtQuantity.Text, intQuantity, ErrorA);
    Val(edtPrice.Text, rlPrice, ErrorB);
    if (edtQuantity.Text = '') and (edtPrice.Text = '') then
        ShowMessage('Quantity and Price cannot both be blank')
    else
        begin
            if (ErrorA <> 0) and (ErrorB <> 0) then
                ShowMessage('Quantity and Price are both be Invalid!! -
Please try again!')
            else if edtQuantity.Text = '' then
                ShowMessage('The edit box for Quantity is blank!! -
Enter a value for Quantity')
            else if edtPrice.Text = '' then
                ShowMessage('The edit box for Price is blank!! - Enter a
value for Price')
            else if ErrorA <> 0 then
                ShowMessage('Please try again! - Invalid Quantity')
            else if ErrorB <> 0 then
                ShowMessage('Please try again! - Invalid Price')
            else if (intQuantity < 0) and (rlPrice < 0) then
                ShowMessage('Quantity and Price cannot both be
negative')
            else
                begin
                    Val(edtQuantity.Text, intQuantity, errorA);
                    Val(edtPrice.Text, rlPrice, errorB);
                    rlCost := intQuantity * rlPrice;

                    rlVat := rlCost*0.175;
                    Str(rlVat:0:2, strVat);
                    edtVat.Text := StrVat;

                    rlGrossCost:= rlVat + rlCost;
                    Str(rlGrossCost:0:2, StrGrossCost);
                    edtGrossCost.Text :=StrGrossCost;

                    Str(rlCost:0:2, strCost);
                    edtNetCost.Text := strCost;
                end
            end
        end
    end
end;

```

```

        rlMemory :=rlMemory + rlGrossCost;
        rlMemoryNetCost := rlMemoryNetCost + rlCost;
        rlMemoryVAT := rlMemoryVAT + rlVAT;
        rlMemoryGrossCost := rlMemoryGrossCost +
rlGrossCost;

        edtQuantity.Text :='';
        edtPrice.Text :='';

        end;

    end;

end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    rlMemory :=0.0;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    rlMemory :=0;
    rlMemoryNetCost:=0;
    rlMemoryVAT :=0;
    rlMemoryGrossCost :=0;
end;

procedure TForm1.btnTransactionClick(Sender: TObject);
var
    strMemory, strMemoryNetCost, strMemoryCash, strMemoryChange,
    strMemoryVAT, strMemoryGrossCost: string;
begin
    str(rlMemoryNetCost:0:2,strMemoryNetCost);
    edtNetCost.Text :=strMemoryNetCost;

    str(rlMemoryVAT:0:2,strMemoryVAT);
    edtVAT.Text :=strMemoryVAT;

    str(rlMemoryGrossCost:0:2,strMemoryGrossCost);
    edtGrossCost.Text :=strMemoryGrossCost;
end;

procedure TForm1.btnClearClick(Sender: TObject);
begin
    edtQuantity.Text :='';
    edtPrice.text :='';
    edtNetCost.Text :='';
    edtVAT.text :='';
    edtGrossCost.Text :='';
    edtCash.Text :='';
    edtChange.Text :='';
end;

procedure TForm1.btnNextClick(Sender: TObject);
begin
    edtQuantity.Clear;
    edtPrice.Clear;
    edtNetCost.Text :='';
    edtVAT.text :='';
    edtGrossCost.Text :='';
    edtCash.Text :='';
    edtChange.Text :='';

```

```

        lblNext.visible :=true;
end;

procedure TForm1.edtQuantityChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.edtPriceChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.edtNetCostChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.edtVATChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.edtGrossCostChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.btnChangeClick(Sender: TObject);
var
    rlGrossCost, rlCash,  rlChange : real;
    strChange : string;
    Error: integer ;
begin
    Val(edtGrossCost.Text, rlGrossCost, Error);
    Val(edtCash.Text, rlCash, Error);

    rlChange := rlCash-rlGrossCost;

    Str(rlchange:0:2,strChange);
    edtChange.Text :=strChange;
    if rlCash < rlGrossCost then
        ShowMessage('Not enough money tendered')
    else
        begin
            lblThank.visible :=true;
            lblAnd.visible :=true;
        end;
end;

procedure TForm1.edtCashChange(Sender: TObject);

```



```
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;

procedure TForm1.btnTotalClick(Sender: TObject);
var
    strMemory : string;
begin
    str(rlMemory:0:2,strMemory);
    edtTotal.Text :=strMemory;
end;

procedure TForm1.edtChangeChange(Sender: TObject);
begin
    lblNext.visible :=false;
    lblThank.visible :=false;
    lblAnd.visible :=false;
end;
end.
```

Screen images of forms used

I have insert two screen images of the form for both Tasks 1 to 5 and Tasks 6 to 8 from Borland Delphi Professional Version 4 using Alt + Print Screen keys.

- Screen image of the form for Tasks 1-5 to meet the level of Pass

The screenshot shows a window titled "Cash Register" with a blue title bar. The main area has a light gray grid background. On the left side, there are five labels: "Quantity", "Price", "Cost (Ex- VAT)", "VAT@17.5", and "Total Cost". To the right of each label is a white rectangular input field. On the right side of the form, there are two buttons: "Enter" and "Transaction".

Saved as A:\HndBit\Isd\Delphin\Assignment 1\Task 5\Cash5.dpr – Cash5.dfm
Cash5.pas

- Screen image of the form for Tasks 6-8 to meet the level of Merit

Cash Register

Books Bargains

Quantity

Price

Net Cost

VAT@17.5

Gross Cost

Cash

Change

Enter Clear

Next Transaction

Change Total

Total Cost (inc. VAT)

**Thank You For Your Custom
And Please Call Again
Next Customer Please**

Saved as A:\HndBit\Isd\Delphin\Assignment 1\Task 8\Cash8.dpr – Cash8.dfm
Cash8.pas

Test Plan for Cash Register Program

Test blank entries:

edtQuantity.Text blank
edtQuantity.Text blank
edtQuantity.Text non-blank
edtQuantity.Text non-blank

edtPrice.Text blank
edtPrice.Text non-blank
edtPrice.Text blank
edtPrice.Text non-blank

Test invalid numeric values:

EdtQuantity.Text numeric
EdtQuantity.Text numeric
EdtQuantity.Text non-numeric
EdtQuantity.Text non-numeric

edtPrice.Text numeric
edtPrice.Text non-numeric
edtPrice.Text numeric
edtPrice.Text non-numeric

Test Number format:

edtQuantity.Text integer
edtQuantity.Text real
edtPrice.Text integer
edtPrice.Text real

ErrorA
ErrorA
ErrorB
ErrorB

Test negative values for quantity and price:

Quantity ≥ 0 , price ≥ 0
Quantity < 0 , price ≥ 0
Quantity ≥ 0 , price < 0
Quantity < 0 , price < 0

positive cost calculated
negative cost calculated
negative cost calculated
error message displayed

- Test validation (using the Enter button)

Quantity	Price	Reason	Net Cost	VAT@17.5	Gross Cost	Total Cost (inc. VAT)	Message
Blank	Blank	Test blank entries	Not changed	Not changed	Not changed	Not changed	Quantity and Price cannot both be blank
blank	Numeric	Testing blank Quantity	Not changed	Not changed	Not changed	Not changed	The edit box for Quantity is blank!! – Enter a value for Quantity
Numeric	Blank	Test blank	Not changed	Not changed	Not changed	Not changed	The edit box for Price is blank!! –

		Price					Enter a value for Price
Jen	Ken	Invalid Quantity & Price	Not changed	Not changed	Not changed	Not changed	Quantity and Price are both be Invalid!! - Please try again!
Jjj	12.3	Invalid Quantity	Not changed	Not changed	Not changed	Not changed	Please try again! - Invalid Quantity
123	Jjj	Invalid Price	Not changed	Not changed	Not changed	Not changed	Please try again! - Invalid Price
1.85	22.99	Invalid real Quantity	Not changed	Not changed	Not changed	Not changed	Please try again! - Invalid Quantity
19	1.85	Fractional Price	35.15	6.15	35.15	41.30	
2	85	Integer Price	170.00	29.75	170.00	199.75	

- Test negative values (without using the transaction button)

Quantity	Price	Reason	Net Cost	VAT@17.5	Gross Cost	Total Cost (inc. VAT)	Message
19	1.85	Negative Price	-35.15	-6.15	-41.30	Not changed	
-2	85	Negative Quantity	-170.00	-29.75	-199.75	Not changed	
-19	-19	Negative Quantity & Price	Not changed	Not changed	Not changed	Not changed	Quantity and Price cannot both be negative

- Test negative values (using the transaction button)

Quantity	Price	Reason	Net Cost	VAT@17.5	Gross Cost	Total Cost (inc. VAT)	Message
19	-1.85	Negative Price	0.00	0.00	0.00	0.00	
-2	85	Negative Quantity	0.00	0.00	0.00	0.00	
-19	-19	Negative Quantity & Price	0.00	0.00	0.00	0.00	

- Test negative values (using the enter and total button)

Quantity	Price	Reason	Net Cost	VAT@17.5	Gross Cost	Total Cost (inc. VAT)	Message
19	-1.85	Negative Price	-35.15	-6.15	0.00	0.00	
-2	85	Negative Quantity	-170.00	-29.75	0.00	0.00	
-19	-19	Negative Quantity & Price	0.00	0.00	0.00	0.00	